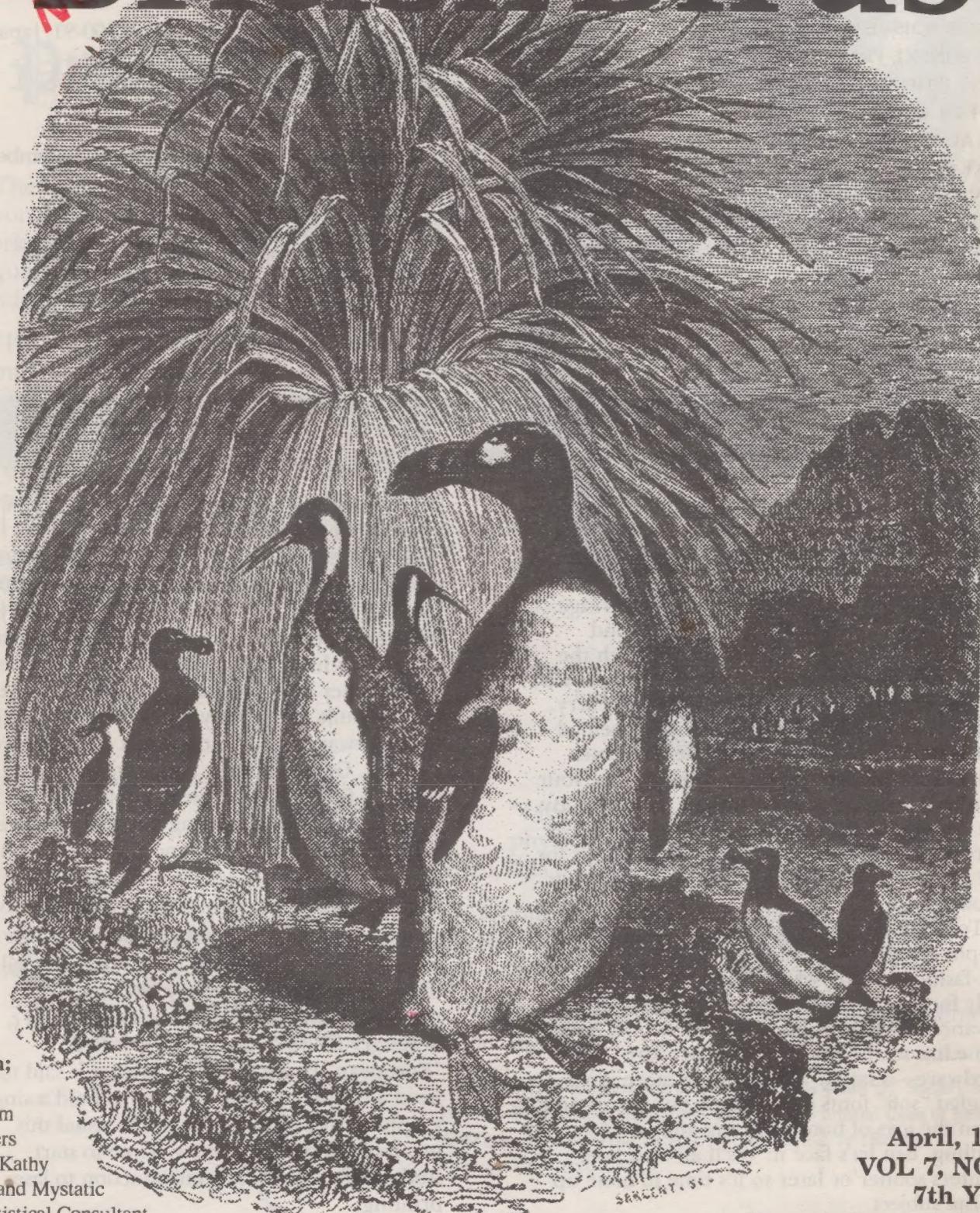


# Olson's Standard Book of British Birds

TOKYO  
Newsletter



Within:

Awk  
Drivparm  
Xywriters  
Prue & Kathy  
Mystat and Mystic  
The Statistical Consultant  
Words through Windows  
Installing those mothers  
Member Profile—Martin Bruczkowski

April, 1990  
VOL 7, NO. 4  
7th Year

Ornithological Issue  
Non-Members ¥300  
Svalbard 13 NKR (allow 1 year for delivery)

Japan's Largest IBM-Standard Personal Computer Users Group

# Tokyo PC Newsletter

Since 1984

ROGER BOISVERT, President

(03) 581-5571

SUE WINSKI, Vice President

(03) 714-4448

TONY WHITMAN, Treasurer/Membership

(03) 349-8521

MATAN ARAZI, Sysop (255)

(03) 582-0785

JIM ADAMS, Librarian

(03) 5562-3703

MIKE GRAY, Newsletter Publisher

(03) 464-8736

DAVID PARRY, Assistant Publisher

(03) 221-0571

PETER EVANS, Newsletter Editor

(03) 728-0728

DAVID BERNAT, Assistant Editor

(03) 5371-3488

Central P.O. Box 1145, Tokyo 100-91, Japan

TPC BBS 24 hours/day

300/1200/2400/9600 baud — 8/N/1

(03) 707-4689

Information hotline for prospective members:

(03) 5706-0057

*The club meeting is held on the first Thursday of each month at the Tokyo Union Church between Omote-Sando Station and Harajuku Station, starting from 7 p.m.*

## March Meeting Highlights

If you want to know more about laser printing, it seems you've come to the right epoch—but it may be over before some of us (present company included) know how to download a font, fish in a bitstream, or rewind a cartridge. The March 1 meeting featured our second introduction to laser printers—with much the same cast of characters as the first held last year: Patrick Hochner, Mike Gray and David Parry. Pitched on an elementary level which seemed about right, the panel fielded queries about fonts, typefaces, points, bits and how to fix your printer when a label gets stuck to the drum. By the end of the meeting, several in the audience were feeling the bite: searching their wallets for the cash needed to buy a little HP or one of its Brothers.

To those unprogressive types content with a 24-pin dot matrix printer (or heaven forbid, an old-fashioned daisywheel that prints beautiful, fully formed letters but doesn't require an advanced degree in paste-up), the whole laser scene has seemed an excuse to procure hardware—a larger hard drive to store new-fangled "soft" fonts on, to name one item—and learn the joys of home kerning and toner refilling. But let's face it: We'll all own laser printers sooner or later so it's time to bone up on the subject.

The panel discussed the relative advantages of Bitstream and Glyphix and the difference

between bit-mapped and scalable (PostScript, for example) fonts—a matter that lends itself to misinterpretation. With a bitmapped typeface, it seems, each type size, from 4 points up to 600, requires its own font in the printer's memory (either on a cartridge or downloaded from the computer's hard disk); with scalable typefaces, on the other hand, an infinite number of sizes may be built. Either way, the word-processing or DTP software has to know about a font in order to use it. Got that? Italic and boldface constitute separate fonts, so any decent publication (such as the Journal currently being perused) requires a healthy dozen or so, plus another slew if you want to add those indecent headlines. If you want to add "pi characters" (the ſ in Jíří for example) or the Č symbol for the yúbinbango or even a little Č logo there is software that will do this—very usefully, as many laser fonts do not include those characters such as ň and ġ, necessary for what are quaintly called "European" languages. The printer is said to need a couple of megs in order to hold a single graphics page in memory, and as usual this means buying extra. Whew. Time to start studying for next year's introduction to laser printing.

On the plus side, laser printers are claimed to run so quietly you can hear your wallet draining.

# Learning to Compute with Prue and Kathy

The February SWET Newsletter carried a report by Prue Moodie and Kathy Ono on a workshop for SWETers "to become better acquainted with computers". Some of the wisdom therein:

"It's easy to be tempted by articles in computer magazines...but do you really need a window feature that lets you look at six documents at once on six pieces of paper?"

"Computer systems have components—monitors (screens), disk drives and 'brains', and printers. If the size of any of these components, or some other aspect, varies from one system to another, the systems will be incompatible."

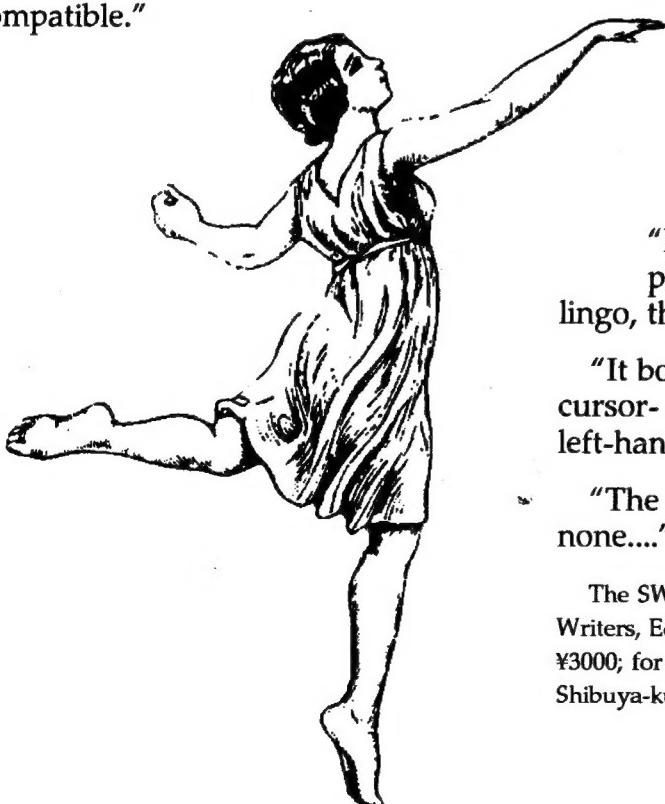


"Modems...can enable you to link two physically incompatible computers (in the lingo, this is called a gender change!)." □

"It boils down to little things like WordStar's cursor-moving keys being on the left, which left-handed people find attractive."

"The usefulness of computer magazines—almost none...."

The SWET Newsletter is just one of the goodies the Society of Writers, Editors and Translators provides for annual dues of just ¥3000; for details, write to SWET at Shibuya 2-19-15-808, Shibuya-ku, Tokyo 150. □



# Awk: A Language for Non-Programmers

by Wm. Auckerman

Writing this article is bringing out my teaching instincts, so in this spirit let's start things off with a multiple-choice question. (There will be another quiz at the end, so pay close attention.)

Awk is:

- (a) A small- to medium-sized sea bird of family *Alcidae*.
- (b) The nickname of this article's author.
- (c) The sound that TPC's sysop utters when the BBS crashes.
- (d) A programming language described by *Byte* magazine as "a powerful and fun-to-use language".

The correct answer, of course, is (d). (And just for the record, answers for the other choices are (a) auk, (b) Auck, and (c) Oh, shucky darn.) Now, you may never have thought of applying the term "fun" to programming, but in this article, I'm going to try to convince you that awk is indeed a rare bird: a full-fledged language that is both attractive enough to seduce even the most avowed programophobics yet powerful enough to satisfy the serious programmers among you.

Do you turn pale at the mention of terms like "object code", "compile time," and "function libraries"? Do your computer buddies snicker uncontrollably when you let it slip that you're still writing in BASIC? Have you ever yearned to write a program that would actually do something useful without spending three weeks chained to your computer? If you answered "yes" to one or more of these conundrums, then read on. (If you didn't, read on anyway.)

---

## A Personal Experience

---

You don't have to be a programming genius to become comfortable with awk. Until last year, my programming successes were confined to a few spaghetti-like, molasses-slow BASIC routines. I had flirted with Pascal and made a serious attempt to decipher C, but after reading a couple of ponderous tomes and bungling the occasional simple programs that I attempted, I gave those up as a lost cause.

Then, on the TPC BBS, I read through a series of messages extolling the virtues of awk. They made awk sound so darn simple and useful that I decided it was worth a try. So I downloaded a shareware version from the BBS, read the meager documentation, and tried my hand at a few small programs. It would make a nice story to say that I was immediately successful, but in fact my first efforts were frustrating—although I managed to accomplish a few simple tasks, I felt as though I was stumbling around in the dark. The documentation was too cryptic, and the shareware version was a pale imitation of the real thing.

A couple of months later, a different shareware version of awk was uploaded to the BBS, so I determined to have another fling. This time, I was more successful, mainly because this version was less buggy and its documentation easier to understand. Spurred on by my successes, I searched out and bought a copy of the 210-page "awk bible", *The AWK Programming Language*, and I quickly became an awk advocate.

---

## A Brief Awk History

---

I know you're all eager to see awk in action, but first let me get all of the background material out of the way.

If you've been wondering about the name, awk was originally designed and implemented in 1977 by Alfred Aho, Peter Weinberger, and Brian Kernighan. (Since the name is an acronym of the creators' initials, I would prefer to write it as AWK, but other printed sources insist on the lowercase form.) According to Messrs. A, W, and K, their goal was to create a UNIX-based pattern-scanning language that was more generalized and flexible than sed and grep. The original awk was primarily for "in-house" use (at AT&T Bell Laboratories), but it quickly spread to "outside" users who stretched the language to do things that its creators had not envisioned. In response to this popularity,

Messrs. A, W, and K released an expanded and enhanced version of awk in 1985.

Long relatively little-known outside of the sheltered halls of UNIX, awk has been adapted to run on MS-DOS computers. Currently, there are several implementations of awk, both shareware versions (*Bawk*, *Dawk*, and *Gawk*) and commercial versions (including *PolyAWK* and *MKS Awk*).

## Brass Tacks

Let's get down to specifics. Awk is a general-purpose programming language suitable for performing routine data-manipulation tasks—especially retrieving, transforming, or validating data; processing and printing text files; creating and managing small personal databases; and crunching numbers.

A typical awk program consists of a series of "patterns" (which tell what to look for in the input data) and "actions" (which tell what to do when the pattern is found). An awk program is generally short and can be stored as a text file or directly input from the DOS command line.

When an awk program is run, data is read from an input file, and each line (record) of data is separated into discrete fields. The record and field separators are user-definable; by default, they are a carriage return and space, respectively.

Awk processes each input line in sequence. If the current input line matches the pattern, the action (which can consist of any number of statements) is executed. If the pattern part of a pattern-action statement is omitted, all input lines are matched, and if the action part is omitted, the default action is to print each matched input line.

## Three Examples

It's impossible to demonstrate the versatility and utility of awk with just a few examples, but I'm going to try anyway. If I present a relatively long sample program of a dozen-or-so lines (yes, 12 lines is long for an awk program), I risk losing the easily daunted among you. So I'll try to confine the examples to short, easily digested tidbits.

### Sample Program 1

We'll start with a "one-liner" entered from the command line. Let's assume that this newsletter article has been saved as an ASCII text file named THIS.TXT. In order to print every line of the file that contains the string TPC, at the DOS prompt we would enter:

```
awk "$0 ~ /TPC/ {print NR, $0}"  
this.txt
```

In the pattern part of this program, \$0 designates a record (by default, a line), ~ means "matches", and the string within the slashes is the item to be matched.

In the action part of the program (enclosed by braces), NR is the (line) record number and \$0 is the record (output line). Yes, this single line is a complete (albeit trivial) awk program. It says "for each line in the file THIS.TXT that matches the string TPC, print that line preceded by the line number." (In other words, we've simulated the DOS command FIND /N "TPC" THIS.TXT.) When an awk program is to be read from a text file rather than entered from the command line, we need to use the option -f to indicate this. Thus, if we had stored the program \$0 ~ /TPC/ {print NR, \$0} in a file named FINDTPC.AWK, we would run the program by entering (at the DOS prompt)

```
awk -f findtpc.awk this.txt
```

### Sample Program 2

Let's try something a bit more ambitious. Suppose we want to count how many times each word is used in the file THIS.TXT. First, we would write the following program, and save it in an ASCII text file called WORDFREQ.AWK.

```
{ gsub(/[,;:!\.\(\)\?\/\|\<\>\+\*\-\]/,  
"")# line 1  
for(i = 1; i <= NF; i++) count[$i]++  
# line 2  
}  
END  
{  
for(w in count)  
print count[w] " ", w # Print each  
word preceded by number of  
occurrences  
}
```

The symbol # indicates a comment; all text following a # on a line is ignored by the awk program.

Line 1 of this program removes punctuation (otherwise, "word" and "word." would be counted separately). It may look complicated, but it simply performs a global substitution, replacing any characters inside the brackets with a null string. (The backslash is necessary in front of some of the punctuation symbols because these are "metacharacters" that have special meanings in awk. The preceding \ preserves the literal meaning of the character.)

Line 2 creates an array to hold the words, and the ++ operator increments the count of each array element. (Note that each array subscript is a string. The result will be such array elements as count[the], count[word], and count[is].)

The END portion of the program, executed after all input lines have been processed, prints each word preceded by the number of times it occurred. (In this simple program, we did nothing to convert uppercase and lowercase characters, so "the" and "The" will be counted separately.) To run the program, we would enter (at the DOS prompt):

```
awk -f wordfreq.awk this.txt
```

The output of this program is in no particular order; it is simply an unsorted listing. However, we could pipe this awk program into a sort program and redirect output to another file, which would produce a listing of words either in alphabetical order or order of frequency. For example:

```
awk -f wordfreq.awk this.txt | sort  
-r -n > wordlist.txt
```

To test sample program 2, I ran it on this text (before final editing) and found that the five most frequently used words in this article are: *the*, 105; *to*, 64; *a*, 63; *of*, 62; *and*, 59.

### Sample Program 3

And finally, a longer example. As a writer, I often want to know how many words or characters are in the magnum opus that I have just completed. This final example is a "bare-bones" version of the program that I wrote for personal use; it will count the number of characters, words, sentences, and lines in an ASCII text file. (My "deluxe" version also counts punctuation, calculates average characters per word and words per sentence,

and uses numerous ANSI escape sequences to format the output.)

First, we'll save the following program to a file called COUNT.AWK.

```
BEGIN { printf("ESC[2J") } # Clear  
screen  
{ wc = wc + NF # Increment count of  
words (default awk fields)  
cc = cc + length($0) + 1 # Increment  
character count (add 1 for CR at  
end of line)  
if(NF == 0) bl++ # Increment count of  
blank lines  
for(i = 1; i <= NF; i++)  
{ if($i ~ /[.!?][)]?$/){ sc++} #  
Increment sentence count (on  
encountering any of ". ? !" the  
end of a word, optionally followed  
by a closing parenthesis or  
quotation mark)  
}  
END  
{ printf("This file contains: \\n%d  
characters, %d words, and %d  
sentences \\n", cc, wc, sc )  
printf("%6d lines (including %d blank  
lines)\\n", NR, bl )  
} # Print results
```

The comment lines explain the purpose of each step. (Note that, without the comments, the program comprises less than ten lines of source code.) In the END portion, \n represents a linefeed and %d indicates that the count is to be printed as a decimal integer.

To run this program on the text file THIS.TXT, we would type

```
awk -f count.awk this.txt
```

The output of this program would be something like

This file contains: 14674 characters, 2387 words,  
and 156 sentences 336 lines (including 82 blank  
lines)

## Where the Awktion Is

For "one-time" applications in which execution speed is less important than program-creation time, awk is ideal. So many activities in awk are automatic (such as input, field splitting,

storage management, and initialization) that an awk program is generally smaller and simpler than the equivalent program in any other language. And because it is an interpreted rather than a compiled language (that is, the awk execution file must be present to run an awk program), revising an awk program is simple.

As you may have noticed from the sample programs, the action language of awk resembles C, but is greatly simplified. (For example, there are no declarations, and strings and numbers are built-in data types.) So if you're already comfortable with C, you'll find the transition to awk easy. And if you want to learn C but have been daunted by its complexity, you'll find awk to be an excellent primer.

Awk actions include many C-like control statements, such as:

```
do statement while (expression)
for (expression; expression;
     expression) statement
if (expression) statement [else
    statement]
while (expression) statement
```

Awk also offers numerous built-in variables and field variables; functions for arithmetic, comparison, and logical operations; string functions (such as index, split, substitute, and substr); multiple input streams; associative arrays; and user-definable functions.

## Some Final Words

Even if you already have a favorite programming language, awk can be a valuable addition to your "programming toolbox". And if you're a novice who would like to learn programming for personal use, I wouldn't hesitate to recommend awk as a "first language": it's just as easy to learn as BASIC, and it provides a much better foundation if you later want to tackle C or Pascal.

If you want to get acquainted with awk before paying for a commercial implementation, try out one of the shareware versions. (You may, like me, even decide that the shareware version satisfies your needs.) Two shareware versions, GAWK202 (Gnu awk) and AWK210 (Duff AWK), are available on the TPC BBS.

My personal recommendation is the "free" shareware version from The Gnu Project: Gnu awk 2.02. *Gawk* conforms closely to the definition and description of the language as presented in *The AWK Programming Language*. (The major exceptions are that it does not permit the use of pipes within a program, and it does not support the awk "system" function. If you do decide to try it, I'll mention that version 2.02 of *Gawk* has a slight bug in its memory management procedures. With very large chunks of data, a "memory allocation error" sometimes occurs. This won't cause any damage to your source or data files, but it may cause the program to abort.)

The documentation with Gnu awk or Duff awk is helpful (but not very). If you are really serious, you'll soon want to purchase a copy of the quintessential source of awk information: *The AWK Programming Language*, by Alfred V. Aho, Brian W. Kernighan and Peter J. Weinberger (Addison-Wesley, 1988, ISBN 0-201-07981-X; available for ¥5,000 at Maruzen or Kinokuniya—there is also a Japanese translation from Toppan for ¥3,400). This relatively slim paperback is terse and to the point. It may take several readings before everything becomes clear, and you'll probably learn more from dissecting the sample programs than from reading the text portions.

## Final Quiz

Okay, I promised another quiz at the end of my article, so here it is. Awk is:

- (a) A seductive, fun-to-use programming language.
- (b) A great language for text processing and data manipulation.
- (c) A powerful, versatile language for expert programmers.
- (d) An easy-to-use language ideal for novice programmers.

If you've been paying attention, you know that the correct answer is "all of the above". □

# Mystat

by Larry Cisar

Have you ever wanted to figure out the standard deviation, skewness, or range of some test scores? Or to find the relationship between price hikes and the income of your company? These are only a few of the statistics that Systat, Inc.'s two programs *Mystat* and *Business Mystat* can quickly calculate.

*Mystat* and *Business Mystat* are subsets of *Systat*. The major differences between the two lie in the range of functions that each offers; the basic set-up is the same. For either program you'll need 512 kB of RAM. No extended memory is needed, no particular type of monitor, not even a hard drive.

The programs each come with a simple manual, which is deceptive in that you do not expect to find so much sound information in less than 30 pages. It begins with detailed instructions on how to copy the programs to another floppy or to install them on your hard disk. Don't let the obviousness of the presentation turn you off, as the manual becomes invaluable when telling you how to run the programs.

When you are in the actual program, you enter your data using the EDIT command. This starts by giving you a blank spreadsheet and having you enter the name and type of each variable. You are limited to 50, which may be numeric or character. For many research projects, this would be a very low number; moreover, variable names are limited to 8 characters. The values entered as data are limited to 12 characters: a problem if you need to enter names as simple as "Setsuko Hayashi". You could set up two variables—one for first names and another for last—but this would use up one extra variable. The number of cases is limited to 32,000. This is more than sufficient for any research I have seen educators working on. Business people might have a different opinion. I for one would have been happy to forgo this

number of cases if this had allowed more variables.

When you are in the EDIT mode you can

When your editor got his paws on the PC Magazine "Inside the World of Numbers" issue (14 March '89) he naturally headed straight for the page on "Shareware and free statistical software". Richard Goldstein there names some good programs (mostly special-purpose), but warns that many are unreliable. He concludes that the best among the general-purpose packages is *Mystat*, a smaller version of *Systat* that is virtually free.

*Systat* seems to win the recommendation of any magazine doing a comparative test of PC statistics packages, finishing very slightly ahead of, or behind, SPSS/PC+. An example is the PC Mag number-crunching issue, where, together with Statgraphics, the two won "Editor's Choice" among programs for advanced statistical analysis. *Systat* for the Mac is a little

create new variables based on the other data you have entered. You do this by hitting Escape, which moves you from editing the file to the command line. You then type in the name for the new variable and the formula for making it. When you hit Enter, *Mystat* creates the variable and does all the calculations. This should be done after all the cases have been entered as *Mystat* will not remember formulae.

The most important command both in and out of EDIT is HELP—not because the programs are difficult to use, but because the help is so good. It tells you exactly what you must do to use the command that you want, and is even better than the fact-filled manual. EDIT only gets you started. Once you have saved your file and exited EDIT, you are presented with a menu of all the statistical analyses you can perform. First, you will need the USE command as *Mystat* will not assume that you want to use the last file you edited. There are graphic, statistical, and

Continued on Page 10

# Mystatic

by David Thompson

On the Mac, *Systat* is widely acknowledged as the only program offering an exhaustive set of statistical methods. Add fancy 3D plotting and

different, and comes complete with Sygraph, available for the IBM either separately or in a package deal. There's also a cheaper program for the Mac called *Fastat*: its manual, says the company, "is easy to read and, better still, easy not to read."

*Systat, Inc.* doesn't allow you or me to distribute *Mystat* or *Business Mystat*. Whether for the PC or the Mac, one copy of each costs just \$5; you stand a good chance of getting one free if you ask politely and on a moderately impressive letterhead. Lucy Saunders, *Systat, Inc.*'s Director of Public Relations, writes to say that airmail to Japan will push the price up to \$20; perhaps a group order would help to bring it down.

*Systat, Inc.*, 1800 Sherman Ave., Evanston, IL 60201  
tel (708) 864-5670, fax (708) 492-3567

"zoom" the View window, that holding graphs produced by the program. Then you notice how the cursor in this window changes to a little hand when you hold down the mouse button and drag. In the Mac's visual language, this hand cursor represents a tool for scrolling a graphic image larger than the window through which it is viewed. But you try to scroll the graphic in the View window with the hand and nothing happens. Strange. Reading the documentation, you learn how to size the graphs. Instead of the standard Mac technique of selecting a graphic object and dragging the handles that appear, you first define a frame by holding down the mouse button and dragging an invisible rectangle. When you generate a graph, it appears scaled to fit within this rectangle. Weird. The hand lets you move the invisible rectangle after you've defined it, but before you can see it. After a graph fills the rectangle, the whole thing's fixed in concrete, absolutely immovable. WYDSIWYG, and truly bizarre.

Bizarro, Pissarro you may think, a Mac's a Mac. What's the difference between Ronald McDonald and Bozo the clown? Well, "consistency" is the obvious answer. It's a pity the creators of *Systat* don't understand the Mac. Writing code that will run on the pedestrian Mac Plus for almost every conceivable statistical test is no mean feat. Writing a concise manual for the five-dollar *Business Mystat* demo and explaining the program in clear, understandable English is nothing less than a minor miracle. Yet by ignoring the conventions that give the Mac its fabled ease of use, programmers Leland Wilkinson and Mark Bjerknes have created a little monster guaranteed to put Mac users ill at ease.

The problem goes deeper than bad cosmetics. *Systat* the program and *Systat* the company carry on a tradition that the Mac was created to

Continued on Page 10

## Mystat

regression commands. While *Mystat* will tell you how to run them, you must know which are most appropriate as there is no explanation of statistics in this program.

PEARSON (CORRELATE in *Business Mystat*) is a good example of a useful command. It will produce a chart showing the correlations between different variables. I found a major mistake in both packages here: the default is listed as "Pair-wise" when it turned out to be "List-wise". This difference leads to totally different results if you have just a little information missing. I did not encounter any other problems like this, but this does not necessarily mean there are none. (A minor problem was that the program did not produce a significance chart to go with correlations.)

As I learned in graduate school, it is important to test your statistics on two different programs to make sure you are getting accurate results. This is true of all statistical packages. □

## Yet More %πΩβΞμα

In the PC Magazine article mentioned elsewhere, Richard Goldstein adds that Gerard Dallal, a biostatistician at Tufts University, has written a number of special-purpose programs that use data files in *Mystat* format. The Public (software) Library lists a program or package by Dallal called *Stat-Sak* that does a lot, and is free; this may or may not be the same.

Gerard Dallal  
53 Beltran St. Malden, MA 02148 □

## Mystatic

end. Based on a model of action that forces the user to choose a mode, then specify data, and finally execute a command, *Business Mystat* will befuddle and frustrate the Mac user, who expects to act upon objects having obvious properties and is used to navigating through the complexities of a program by moving from event to event. If you know which statistical tests you want to perform on your data and also know exactly how to run them, you can undoubtedly follow the solicitation of the manual's introduction and "...use *Business Mystat* to solve real problems."

If your goal is to learn business statistics, you'd better find a teacher. True to its word, *Systat* has "geared *Business Mystat* to teaching...." Don't expect to learn by using this program on your own. If it's heuristics you want, take a look at *JMP* from the SAS Institute. A beer-budget version (student version officially) sells for \$89 and differs from the \$695 champagne version only in limiting the number of data cells to 500 for a single analysis. *JMP* organizes its tests into six logical suites that it calls platforms (Distribution of Ys, Fit Y by X, Fit Y by Xs, Specify Model, Spin, and Ys by Ys). Given the type of data (ordinal, interval, or nominal) and a platform, *JMP* selects the appropriate statistics and displays the resulting plots in separate windows. You then act upon these graphic objects to further explore your data visually and perform other tests interactively.

The comparison isn't entirely fair, since *JMP*, like most statistics programs on the Mac, emphasizes exploratory data analysis (see "How the other half crunches" on page 16). *Systat* never promises more than a cruncher. If you're content to see your computer as a calculating engine, by all means use *Business Mystat*, and don't bother to read the rest of this article. But if you understand that the computer, even in its embryonic stage, holds the potential to be a universal machine, then finding a program to make statistics less abstruse is probably the least of your problems. Take a long hard look at the beast on top of your desk. XT? AT? No way. 386 running *Windows*? Won't take you any place the Mac hasn't already been. Mac? You don't need a weatherman to know which way the wind blows. □

# The Statistical Consultant by Robert Sechrist

is a free program that suggests the most appropriate statistical procedure for your purpose. The program starts by asking you the number and nature of the variables. It then proceeds with other straightforward questions. At the end, it tells you what you might do, and where you will find an explanation of the technique.

Because the program is an authorized implementation of a book that's almost ten years old—*A Guide for Selecting Statistical Techniques for Analyzing Social Science Data*, 2nd ed., by Frank M. Andrews, et al., which I haven't seen—its tips on software are of little use (*SPSS* and *SAS* are in the bibliography, but in versions that predate the IBM personal computer). However, it works as a good index to the printed statistical literature.

*The Statistical Consultant's* .DOC file includes a lengthy glossary, but this and the program itself assumes that you do already have some training in statistics under your belt.

Here's an example:

Two Variable Treatments

Treatments for one Interval and one Nominal Variable

Is the interval variable dependent? (y/n): y

Do you want a measure of the strength of relationship between the variables (or test the statistical significance of differences between groups) (y/n): n

You have selected to test significance. Are you willing to assume that the intervally scaled variables are normally distributed in the population (y/n): y

Do you want to test the equality of means (or of variances of the dependent variable for different categories of the independent variable) (y/n): y

You have selected to test the equality of means. Do you want to assume homoscedasticity across levels of the independent variable (y/n): n

Suggested Statistical Measure —Analysis of variance

Suggested Statistical Test —Welch statistic, Brown-Forsythe statistic, t test

Reference Hays 1973, 404, 457; Brown & Forsythe 1974a

See Osiris (ANOVA) SPSS (ANOVA, ONEWAY, BREAKDOWN, MANOVA, T-TEST), SAS (GLM, ANOVA, T-TEST)

Common assumptions for inferences based on techniques using one or more intervally scaled variables (particularly when the intervally scaled variable is a dependent variable) include the following: 1), that the observations are independent, i.e. the selection of one case for inclusion in the sample does not affect the chances of any other case

being included, and the value of a variable for one case in no way affects the value of the variable for any other case; 2), that the observations are drawn from a population normally distributed on the intervally scaled variable(s); and 3), if more than one variable is involved, that intervally scaled variable(s) have equal variance within categories of the other variable(s), i.e. there is homogeneity of variance. Bivariate or multivariate normality is also sometimes assumed.

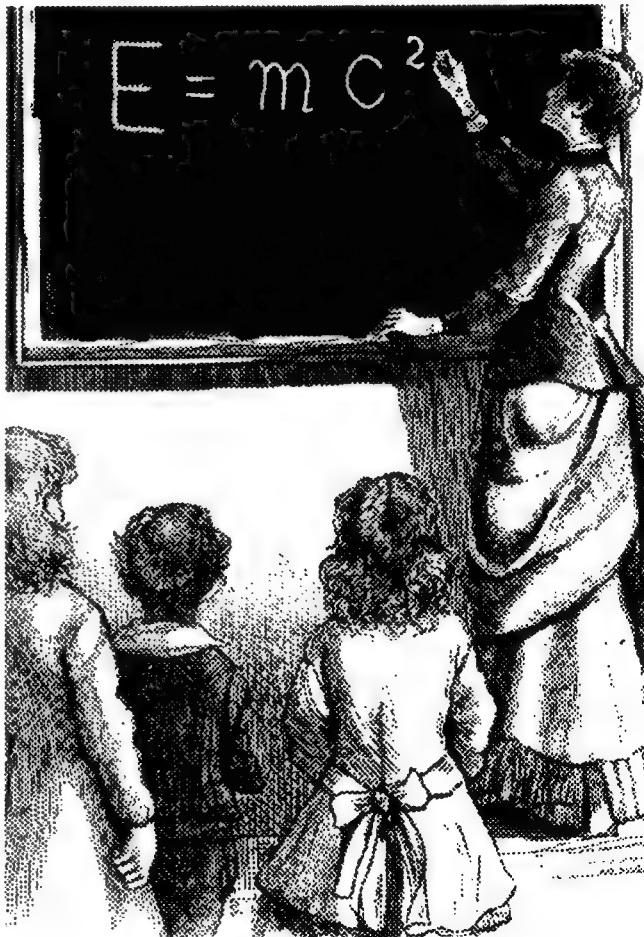
The program's .DOC file also carries a long list of references, which explain "Brown & Forsythe 1974a" and the like.

The program works straightforwardly and the price is right. I wish it made less use of the Hercules' intense mode, but this is a minor gripe.

*The Statistical Consultant* seems not to be carried by some of the usual shareware suppliers—e.g. it's not available from the Public (software) Library—but it is on the TPC and other bulletin boards. The modem-shy might send a few dollars for a disk to:

Robert Sechrist

Dept. of Geography & Regional Planning  
Indiana University of Pennsylvania  
Indiana, PA 15705 □



# DOS Tips for Beginners

## (No. 332 of a Series)

by Mike Gray

DRIVPARM is in effect an installable device driver that allows you to reconfigure your diskette drives in much the same way as DRIVER.SYS, discussed last month, but with a few differences. Instead of *adding* a logical drive (having your drive A: double as drive F: when it is in the 360kB mode) it tells DOS that, say, what was your 1.2 MB drive A: is now a 720 kB drive A: *instead*. This may be handy for those who use Japanese domestic machines and want to retain a greater compatibility with US versions.

The statement you would put in your CONFIG.SYS would be:

```
DRIVPARM = /d:number /f:factor  
          /h:heads /s:sectors /t:tracks /i  
          /n /c
```

The command line switches are similar to those for DRIVER.SYS, or:

d is the number of the physical drive (A=0, B=1, etc.)

h is the number of heads (2 for a floppy)

s is the number of sectors (9 is standard for 360/720 kB, 15 for 1.2/1.44 MB)

t is the number of tracks (80 on a 720 kB, 1.2 or 1.44 MB, 40 on a 360 kB)

i specifies a drive in a system whose standard BIOS does not support 3½ in. drives

c specifies that the drive will know if the disks are changed (all except 360 kB)

n specifies a non-removable device (hard disk)

f is the form factor or standard drive type.

If you specify /f for a floppy there is no need to specify /h or /s or /t.

Valid types are:

0 = 160/180kB single or 320/360 kB double-sided

1 = 1.2 MB (usually 5¼ in.)

2 = 720 kB (usually 3½ in.)

5 = hard disk

6 = tape drive

7 = 1.44 MB (3½ in.)

So if you had a Toshiba with a 40 MB 20 track 40 sector single-head tape backup unit attached to the external floppy port and you always wanted to recognize drive A: as 720 kB you would add the following lines to your CONFIG.SYS:

```
drivparm = /d:0 /f:2 /c
```

```
drivparm = /d:1 /f:6 /h:1 /t:20 /s:40
```

One word of warning, however. Not all versions of DOS support this command. It first appeared in MS-DOS 3.1 but went undocumented. It then disappeared from MS-DOS 3.2 and re-emerged in 3.3. It remains there in 4.0 and 4.01 and is now documented. But beware, many of the flavors of DOS (IBM, Toshiba, NEC, etc.) do not support DRIVPARM. If you are using one of these, try substituting MS-DOS's COMMAND.COM, MSDOS.SYS and IO.SYS for the versions you have been using.

If you are experimenting with a hard disk, remember to keep a "vanilla" DOS disk handy—just in case. And always, always keep a backup of your hard disk!

The /i and /n switches are only for MS-DOS 4.0 and later, and some newer versions of Compaq MS-DOS 3.3.

There is available on the TPC BBS a program by Daniel Fandrich called DRIVPARM.COM which claims to allow versions of DOS that do not support the DOS DRIVPARM command to allow just such support.

It is worth mentioning that due to physical and electrical differences a Japanese 1.2 MB 3.5" drive such as that in the Toshiba Dynabook *cannot* be configured to read both 1.2 MB and 1.44 MB formats.

[Reminder: If your main interest in original approaches to floppies is merely to get as much onto them as common sense and safety (rather than IBM) allow, there are low-cost and free utilities for this. MAXI Form, by Herne Data Systems, costs just \$10. Christoph Hochstätter's FDFORMAT, which crams 1.72 MB onto a 3½ in. HD floppy, is free, and is described by Chris Witmer in "A Shoehorn for Your Floppies" in the December issue. You'll find both of these on your friendly local BBS, and maybe also in the TPC disk library.] □

# XyWriters

will be happy to learn that XyQuest, Inc. is supplying its "Application Notes" and related files free in Japan, via P. & A. BBS.

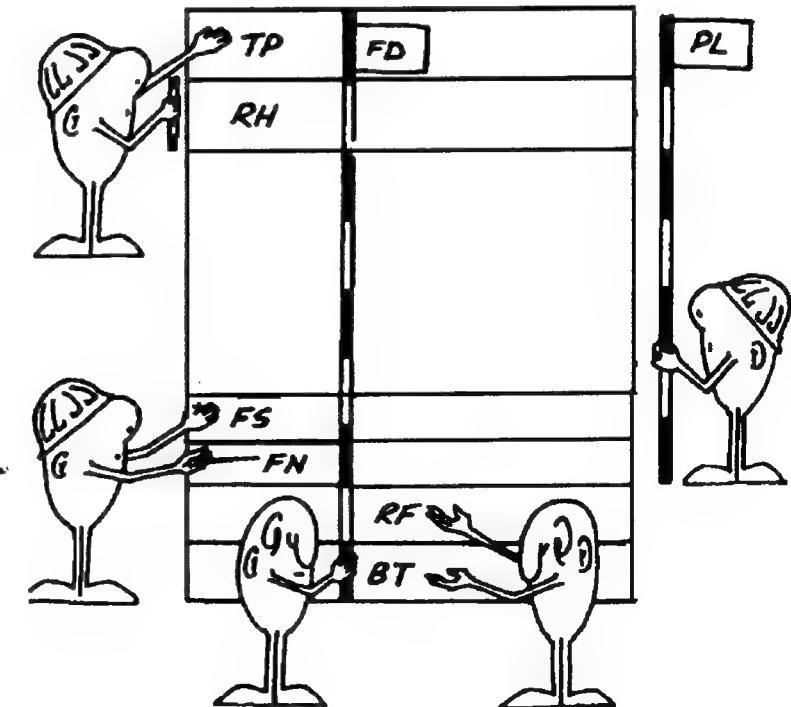
If you are a registered user of XyWrite you probably already know what the Application Notes are. Briefly, they're a series of leaflets each of which explains a particular aspect of the program. Some are elementary; when you buy an upgrade to the program you receive a Note that tells you how to copy the contents of the floppy onto another floppy or a hard disk. Others go into the minutiae of particular printers, or talk about problems with troublesome TSRs. A few are substantial: number 168, "Using XyWrite III Plus with Ventura Publisher 2.0", runs to seven two-column pages and is accompanied by five files.

If you contact XyQuest and ask about a problem that's answered in one of the Application Notes, the company will usually mail you a copy, free, together with a disk containing the necessary software. Alternatively, you can buy a year's subscription, and receive the whole set plus, every quarter, an envelope-full of the new or revised ones. Or again, you can buy a subscription to XyQuest's own BBS and download what you want from there. Now, however, you can download from a Tokyo BBS whichever of these Application Notes or files you want, thanks to an agreement that P. & A. BBS sysop and keen xywriter Patrick Hochner has worked out with XyQuest.

This agreement—one in which no money changes hands, incidentally—prohibits any further distribution of any of XyQuest's material. You may not give out (let alone sell) copies of what's on P. & A. BBS; neither may you upload it on any other BBS. If you or a friend wants something from P. & A. BBS, then you, he or she has to download it directly. (No problem if you have a modem; if you still don't, this could be a good reason to get one.)

XyQuest does not provide all its supplementary material free in this way. P. & A. BBS can't supply you with the many little utilities that you can get free from XyQuest. And the XyWrite Programming Language User's Guide and Word For Word for XyWrite, for example, are nowhere available merely for the cost of a phone call. However, the Application Notes are not all that P. & A. BBS offers to its callers. The board is a center for heavy XyWrite users, generous with their tips, news, lore, programs, files, and confessions. If you're struggling with the world's wildest word processor, or even if you've tamed it, P. & A. BBS is the place to be.

P. & A. BBS (24 hours, no charge,  
1200/2400/9600 bps) 0425 46 9143 □



# Words through Windows

## Part One

by Charles P. Fischer

Because of the complexity and richness of both currently available word processing packages for *Windows*, this article will come in three parts. Here I will present a brief overview of features and specifications of *Ami Professional* and *Word for Windows* and make some generic recommendations. Next month, I'll describe the programs in action and say how they differ in design philosophy. Finally, I'll detail the advanced options of both, particularly the built-in macro programming and custom configuration features.

A Swiss army knife or a professional toolkit? If you're on a budget, have limited memory, hate juggling several tools at the same time, and need writing, desktop publication, and presentation tools tightly integrated in a single package, then *Ami Professional* is for you. If you're already a *Windows* power user, have *Excel*, *Corel Draw*, and *PageMaker* and require speed and outlining power for writing, then you will probably be better off with *Word for Windows*. Naturally, many people fall somewhere between those two extremes, so I hope that this three-part article will help you decide which program is best for you. If, as is likely, you are currently using a DOS text-based word processor such as *WordPerfect*, *Word*, *XyWrite*, or *WordStar*, you will first have to decide whether you want to use any *Windows*-based word processor at all, or just stick to your present package.

Basically, the more formatting control you want, the more *Windows*-oriented you should be. If on the other hand you usually submit "raw" text to your publisher for final formatting and printing, leaving your familiar DOS word processor will not be necessary.

I believe several trends are working to induce more people to switch to *Windows*-based word processors, especially the recent proliferation of low-priced "personal" laser printers (e.g. the HP IIP). With 300 dpi output now more widely available, more people want to have formatting

control of their documents, whether for business or private communication. At the same time, VGA and Super VGA screen resolutions, combined with larger monitors, are making page proofing a viable on-screen option, using bit-mapped screen fonts that correspond to the laser printer fonts. The recent reduction in the price of fast 286 and low-end 386 computers means that more users can afford the minimum practical configuration for these products: basically a 286 or 386 machine running at 16 MHz, a 40 MB hard disk, a mouse, an EGA or VGA monitor, and 2 MB of memory.

The integration capabilities of *Windows*, allowing documents to incorporate graphs, tables of data, and images are enabling users to produce more information-rich letters, memos, proposals, presentations, and reports. Finally, the ease with which ordinary users learn *Windows* products allows them to perform powerful formatting processes on the screen, including style sheets, tabs, columns, headers, footnotes, footers, lists, indexes, and tables of contents. These are generated through a combination of dialog boxes, icons, the ruler, menu choices, and speed keys, including customized macros.

The following table summarizes the features of these two packages. Next month, I will describe how these features actually work when creating documents. □

# Amí vs Word for Windows

<i>Amí Professional</i>	<i>Both</i>	<i>Word for Windows</i>
	<b>Runtime Windows 2.11; laser fonts available/supplied</b>	Discount coupons for Bitstream Fontware 3.0 (\$25) and Agfa Type Director (\$39)
3.5" and 5.25" disks included		3.5" or 5.25" disks available
Quick reference card	<b>Users' guide, reference manual and pre-defined style sheets</b>	Sampler template files and documentation, pocket guide, printer guide, keyboard, mouse and function key templates
On-line macro language help	<b>On-line help</b>	On-line tutorial (CBT)
Free macro language manual available	<b>Sample macros and basic documentation</b>	TECHREF.DOC file contains abridged macro language documentation; full manual available for \$20
British English, French, German, Italian, Dutch and Swiss dictionaries by order; 40,000 root thesaurus	<b>Spelling checker (130,000 words, American English), user dictionary</b>	International spelling pack (\$75) includes Br. Eng., Fr., Ger., Sp., It., Swedish and Dutch; multiple user dictionaries
<i>Advance Write, DIF, Navy DIF, DCA/FFT, Enable 1.5-2.15, Samna Word, Peachtext, WordStar 2000, WordStar 3.0</i>	<b>File conversion: ASCII, Write, DCA/RFT, Multimate Advantage II, WordPerfect 4.2, 5.0, WordStar 3.3,4.0.</b>	<i>Excel (BIFF), MS Word 3.0-5.0, MS Works, Multiplan, RTF, Text (PC-8), WordPerfect 4.1, WordStar 3.4</i>
1-4 levels of undo (user set)	<b>Undo</b>	
"Enlarged" view for detail work "working view" as percent of "standard"	<b>Draft/layout mode toggle, WYSIWYG editing views; print preview</b>	"Normal" view (part WYSIWYG); partially editable print preview
	<b>Outline style sheets</b>	Outline view with full, object-oriented outliner
		Multiple open windows (max 9)
Column guides, individual "frames" of text	<b>Snaking and side-by-side columns; table tool with word wrap inside expandable cells</b>	
	<b>Mail merge and address labels</b>	
	<b>Table of contents and indexing</b>	
Vertical ruler, selectable icons (up to 10 out of 30 pre-defined and 9 user-defined)	<b>Screen configuration options</b>	Ribbon, ruler, status bar
Vector-based drawing and graphics editing tools	<b>Lines and boxes</b>	
Built-in graph tools	<b>Graphics, graph and DDE through clipboard</b>	
PCX, TIFF, Windows Metafile; graphics rotation for bit maps	<b>Graphics import, moving and scaling</b>	TIFF picture files

## How the other half crunches

Exploratory data analysis, as offered by JMP, is a relatively new way of doing statistics. It got its start at the Stanford Linear Accelerator Center in the early 70s with something called PRIM-9, a computer and graphics display system for looking at data in up to nine dimensions. A team led by John Tukey developed PRIM-9 to quickly find graphical evidence of new particles generated by atomic and subatomic collisions. Besides specialized graphics hardware costing hundreds of thousands of dollars, running PRIM-9 required a dedicated IBM mainframe that was among the largest and fastest of its era. The computer time alone for PRIM-9 cost several thousand dollars per hour.

The first step in traditional statistics is to form a testable hypothesis. An experiment is then designed and performed, and data gathered. Statistical tests tell the experimenter whether the data collected support or contradict the original hypothesis. The real world, however, is often more complex. Exploratory data analysis is a way of dealing with the masses of empirical and historical data that exist in fields where controlled experiments are impossible. Paul Velleman, a professor at Cornell and one of Tukey's students, outlines the criteria for an exploratory data analysis package:

- Integration: threads of investigation should be able to follow any path at any time
- Point of view: data is most revealing when it can be viewed from all possible angles
- Comparison: one "what-if" analysis presupposes another; you must be able to compare analyses side by side
- Pipes: the results of one analysis must be available for input to another
- Real-time graphics: immediate interaction and response are essential

All exploratory data analysis packages are based on 3D plots and have tools for rotating the data cloud and skewing the perspective. Other tools include brushes for coloring points and subsets of points, pens for drawing lines between points and creating color surfaces, and, of course, traditional statistical tests. (EDA is the acronym, but we suffer from acronym poisoning, and EDA can also refer to software for Electronic Design Automation.)

David Thompson

## The View from the Lab



Akira Kihara, protozoologist and Macspert, also took a look at *Mystat* and *Business Mystat* for the Mac.

His opinion in brief is that *Mystat* is very rough, and that *Business Mystat*, though smoother, does not make proper use of the Mac interface. For example when you are cutting and pasting within the program's spreadsheet, the undo function does not work. However, the program does have its strengths. It's easy to import or export data, and the graph functions are easy to use and versatile.

Mr Kihara's conclusion:

If you don't already have a statistics program and suddenly need one—at the price they're asking for *Mystat*, you can't lose. □

# Installing those mothers

## (or Only wimps buy computers ready made)

*Dr Cerenkov shows power users how to cram the RAM and set up their motherboards.*

Stage 1—Ground yourself, discharging static electricity. At least touch the metal casing first, before you touch the board(s).

Stage 2—Install those memory chips... *carefully!* Take your time—allow at least thirty minutes for 36 chips. I use a pair of radio pliers to scrape the oxide layer off the legs of these DRAM chips and straighten them out. Make triply sure—yes, check three times—that the chips are correctly orientated, the front notch matching the position of the notch indicated on the silk screen on the motherboard. Insert, making sure each leg makes good contact with the metal grips in the socket. (One user had the back legs of his chips not making good contact, and I had to re-seat several of his chips—now all is fine.)

Stage 3—Connect your standard IBM (clone) power supply connectors—there are two of them—to the motherboard. Your PC/XT/AT power supply must be IBM compatible—don't force connections that obviously don't match! The power supply should be at least 200 watts. (The supply for the original IBM PC delivered only 70W, just enough to say "Hello".) There are two connectors; they should be colored and have notches. Insert the correct way: the one with several red and several black wires goes into the connector nearer to the center of the mother board; the more colorful one connects to the connector nearer the edge—near the keyboard's contacts.

Stage 4—Install only the graphics card and the speaker. Now power up. Wait a few seconds.... Is there a message on the screen? Is the memory counting up? If no, then is the speaker beeping? If so, there's some error—see the manual for what it might mean, and *check your memory!* Any burning hot chips? Maybe they're plugged in the wrong way or you've got bad contacts. If you do get a message, then let's continue.

Stage 5—Turn the power off and connect your AT compatible keyboard. Power up again. Can you set it up? Configure the system. Play around—*now*, not later when your hard disk is in.

Stage 6—It's time to install that hard disk/floppy controller board. Look at the results. Does CHDKSK like your floppy disks? Can you format, write, and read OK? And what about your hard disk partitions? Can you boot from the hard disk OK? Power down and then up a few times—but allow at least 30 seconds before switching on again.

Stage 7—if so far *all* is OK, then now install your other boards. (But attempt to install other memory boards and you're asking for trouble. Extended memory boards may fit in, but avoid the expanded.)

"Back late from the TPC meeting again"



Have you got the hint?

Yes, it's one step at a time! And play around without the danger of losing anything early on. Please experiment, especially if (as with the Marstech boards) the ROM BIOS contains a full complement of set up options. Do not just stick all your boards in, and expect it all to work. I never do, so why should you? Unless of course you want to waste more time in the long run, starting the very next day. A great mother should be treated with some respect! □

# Welcome

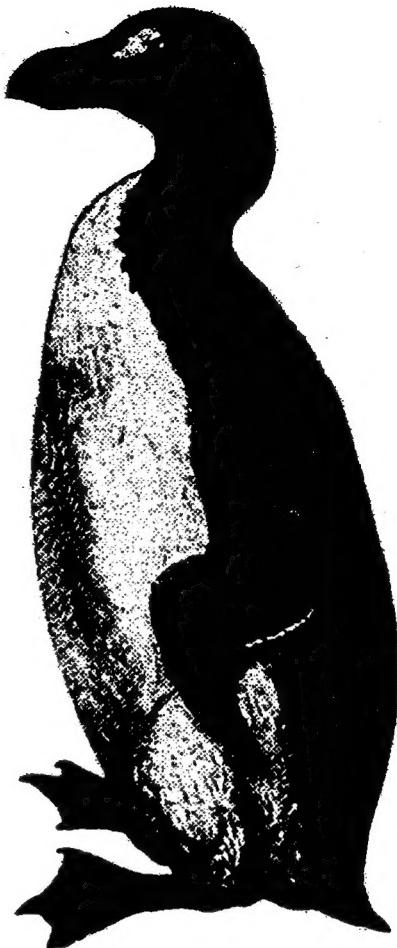
The Tokyo PC Club is pleased to announce that since March 3, 1989, our membership has increased by 17 (all active) to a total of 111 members. To all we extend a hearty welcome. Whether you are a newcomer to PCs or an old hand, we are confident that between discussions with other members at the monthly meetings and on our provocative Bulletin Board, and our world-renowned Newsletter you will find it well worth the expense. All this and our promising software library too!

## Member      Date of joining

P. Bai Akridge	10/05/89
Robert M. Burnside	09/07/89
Peter Colenberg	06/01/89
Naoyuki Fukushima	10/09/89
Shigeo Gomi	08/28/89
Don Green	09/07/89
Bradley W. Homan	10/05/89
Joichi Ito	09/20/89
Dan Kanagy	11/09/89
Steven C. King	12/07/89
Koichi Kobayashi	07/06/89
Thomas Miles	01/11/90
Shigeru Oshio	12/07/89
Richard Reimer	01/11/90
Taylor H. Smissen	11/09/89
Arthur Tanaka	06/15/89
David W. Winchester	01/11/90

# Coming soon!

Database management is a tremendously complex task, and programs for this are necessarily monstrous—but no, says Marty Davidson, this isn't so (or at least needn't be). Marty is a user but not a pusher of *Paradox* (see back page of the last issue). On 5 April he will preside over and lead a soirée intended mainly for the would-be and novice manager of data. Time and place are the same as always. Be there!



The newsletter's "page one bird", surrounded by a bevy of buxom beauties, is the great auk. We don't know where the illustration originally appeared; we scanned it from Jim Harter's *Animals*. This book belongs to the Dover Pictorial Archive Series, as does Harter's *Women*, which we used elsewhere. The many books in the Series offer artwork that's in the public domain (some books from a single source, others from many), although their compilation is copyrighted. Dover allow up to ten cuts from any one book to be used in a project or publication free, and do not demand a credit line (but would appreciate it). Though the Dover Pictorial Archive Series is short on the staples of disk-based click art—pencil-twirling leggy secretaries and the like—it offers a lot. Dover have recently given up sewing their paperbacks, but the paper quality is still excellent and the prices still low (my copy of *Animals*, with 1419 illustrations on 279 pages, cost \$9.95). You will find many Dover books in large English-language bookstores such as Maruzen and arty establishments such as On Sundays.

# "Babbitoons"

© 1990 by A. Colville & D. Kach



## Box this!

Every bulletin board is marked with the personality of its sysop, and TPC's has been no exception. Whenever the hapless have sought succor, wherever the witless have pontificated with foot in mouth, up has popped Maynard Hogg with diagnosis for hardware, software or psyche (never had so many been Joe King). Behind the scenes he has been hard at work, too: validating, promoting, demoting, expunging, renaming, shunting, concatenating, defragmenting, resetting, and (not least) backing up. All this for no pay, and over three years.

Alas, his recent World Tour ("Maynard Does Barcelona", etc.) reminded Maynard of the delights he'd been missing while glued to the screen. A few days back on the board, and he called it a day.

Gokurousama Maynard.

Sprinters, superusers, awk-fanciers rejoice—he's still with us. Welcome back, Citizen Maynard. □

## Laurels and Hardy Perennials

Marty Davidson (sharp-eared mole)  
Tony Whitman (envelopes, labels, hauling boxes around)  
Pablo Muller and his staff (as always)  
Lissa C. Boxy and Samna Corporation (*Ami Professional*)  
Lucy Saunders and Systat, Inc. (*Mystat* times four)  
Paula Lagasse and XyQuest, Inc. (*XyWrite* logo and graphic)  
Dover Publications, Inc. (assorted birds and cranes)

This newsletter contains graphics that are copyright Python (Monty) Productions (probably), and Tom Philbrook, XyQuest, Inc. and Novell, Inc. (definitely). □

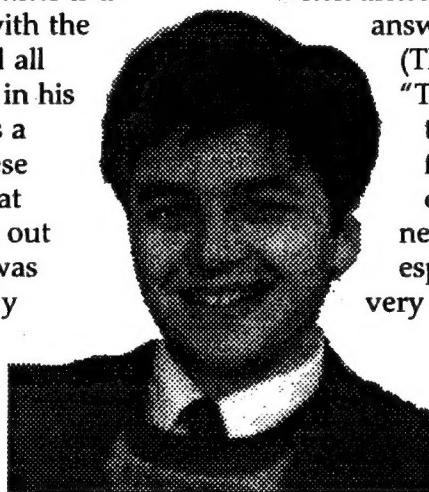
# Member Profile

by Sue Winski

He leans back in his chair with a slightly scampish smile, says "I'm all yours" ...and the interview begins. Martin came to Japan three years ago from Warsaw, ostensibly to study at Sophia University, but there was another, stronger, reason—*cherchez la femme*. La femme, his recent bride, Kazuko. Martin, just 25 on March 5th, has one sister, 9, and two brothers, 13 and 6. His mother worked as an English-French-Polish translator, and his father is a successful musician. As violinist with the Warsaw Philharmonic, he traveled all over the world, and now he plays in his own string quartet. And how does a nice Polish boy meet a nice Japanese girl—and in Warsaw? It was love at first sight. He noticed her walking out the door of his music school and was immediately captivated. He quickly found a place to set the double bass he was holding, and ran out the door to find her. She had disappeared. Spotting her at the bus stop he ran over to her and began chatting. A concert pianist, she had come to Poland for the world-renowned Chopin

Competition, which takes place every five years in Warsaw. Martin worked during the contest as an interpreter. Less than three weeks later he and Kazuko were engaged. After the competition, she stayed on for a while with Martin's family and performed with his father's quartet in concerts around the country.

Electronics has been Martin's hobby since he was young, when he made the amplifiers on the stage speakers for his friends who had a rock group. He was also called on to repair TV sets of family and friends. Together with a friend he bought his first computer, a Sinclair Spectrum, with 1K of RAM and learned BASIC programming. Later he was overjoyed to upgrade to 16K. He doesn't see anything remarkable in the fact that his work, as



**Martin Bruczkowski**

computer hardware technician and now software specialist, is so far from his studies in language and literature. In fact, he always knew he would do something different from his education. After all, "education is learning to think and study, and not a lifetime job." Anyway, linguistics is really not so far off from computer-dom, as math and computers are now used to understand language.

When asked what he liked best about Japan, he answered without hesitation, "TPC".

(This is true, readers, no propaganda.) "The BBS mentally saved me." Even though friendships were first filtered through the modem, the club served an important social need at the time. "For foreigners, especially if you are not American, it's very difficult to find friends. Tokyo is tiring, people are cold, and I sometimes feel tense and depressed." So that's what was happening one day many months ago when this writer signed on to the board to find an undecipherable message (in Polish) followed by the question

"Can you read this?"

And dreams.... "I haven't really thought of it, but off the top of my head...I would like to have two children, live in Poland, own my own computer company (hardware and software), have a house with a Japanese tatami room and serve real Japanese cuisine to our Polish guests. And in one room, I'd like to have my own 'Computer Museum' filled with my collection of computer antiques and oddities." He has already started accumulating the computers. Martin hasn't been back to Poland since he came to Japan three and a half years ago. "I left the Polish People's Republic. I'll go back to the Republic of Poland. The eagle finally has its crown back," he beamed. (The communists removed the eagle's crown in 1945.) □